CLAIMS

1. A method of generating logic control units for railroad Station-based Vital Computer Apparatuses, i.e. for railroad station system control units comprising at least one vital computer which, on the basis of a control program operating in combination with a logic unit, sends state switching controls to so-called yard elements, i.e. devices that are designed to perform specific train circulation-related operations, such as signaling devices and/or railroad switches and/or track circuits, or the like, and receives state feedback and/or diagnostic signals from said yard elements, said logic unit being generated automatically by a program, on the basis of the surrounding conditions as defined by the station diagram, comprising the list of yard elements, and by a state table, wherein state assuming and/or state switching rules are settled for said yard elements, with reference to state and/or to state switching of the other yard elements and/or to the proper management of railroad traffic, said logic unit being a network of circuits with components operating according to Boolean logic functions and appropriately structured in compliance with the station diagram and with the state table, or said logic control unit being a program which includes algorithms composed of Boolean logic functions, which operate like networks of Boolean logic circuits, characterized in that it includes a

step for checking the correctness of the automatically generated logic unit, which checking step includes the following steps:

parallel generation of two logic control units, according to the same station diagram and the same state table, each unit being generated by one of the two generation programs which are as different as possible from each other;

comparison between the networks of logic circuits or the network-simulating logic programs provided the two different generation programs to check for structural differences therebetween.

2. A method as claimed in claim 1, characterized in that, when an identity result is achieved, the correctness of the networks of logic circuits or of the generated logic program is deemed to be checked.

3. A method as claimed in claim 1 or 2, characterized in that, when the two logic programs are found to be non-identical, an error checking step is performed, and the steps of parallel generation of the networks of logic circuits and/or network simulating virtual logic programs are repeated.

4. A method as claimed in one or more of the preceding claims, characterized in that the difference between the two generation programs relates to their languages or to the programming environments wherein they were written.

5. A method as claimed in one or more of the

preceding claims, characterized in that the two different generation programs use different generation algorithms.

6. A method as claimed in one or more of the preceding claims, characterized in that the two different generation programs are two different neural networks.

7. A method as claimed in one or more of the preceding claims, characterized in that it includes a step for preparing a knowledge base containing station diagram related data and state table related data which are coded in such a manner as to be discernible by both generation programs.

8. A method as claimed in claim 7, characterized in that one or both generation programs include a pre-generation step, in which the knowledge base data is checked for consistency and correctness of both data structure and meaning.

9. A method as claimed in one or more of the preceding claims, characterized in that it includes a program for comparing the logic programs and/or the networks of logic circuits generated by the two generation programs, which comparison program is separated from the generation programs.

10. A method as claimed in one or more of the preceding claims, characterized in that the two generation programs generate the logic programs with the following procedure:

Generation of networks of logic circuits which use logic hardware components;

Conversion of the networks of logic circuits so generated into logic algorithms composed of sets of Boolean equations whose behavior corresponds to that of said networks of logic circuits.

11. A method as claimed in one or more of the preceding claims, characterized in that it is used when logic circuits and/or logic programs are to be changed to be adapted to changes of the station system diagram and/or of the state table.

12. A Vital Computer Stationary Apparatus including a computer wherein a program is loaded to control and monitor yard elements of a station system, which operate according to different rules, characterized in that the control program includes a section of general procedure-oriented programs, that are applicable both to the station system structure and to the state table, which program is interfaced and integrated with a control and monitoring logic program, which incorporates the station system structure and the state table, and is automatically generated and checked by a section of the Vital Computer Stationary Apparatus, that may be recalled at will with a method as claimed in one or more of claims 1 to 11.

13. A Vital Computer Stationary Apparatus as claimed in claim 12, characterized in that section for generating the control and monitoring logic program

constitutes a section for changing and/or updating said control and monitoring logic program.

14. A Station-based Vital Computer Apparatus as claimed in claim 12 or 13, characterized in that the section for generating the control and monitoring logic program comprises at least two different generation programs, for generating comparable control and monitoring logic programs which are loaded, after a successful identity check, in the memory of the Vital Computer Stationary Apparatus and are interfaced with the section of general procedure-oriented programs.